

Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

“Trông người lại ngắm đến ta.”

— Tục ngữ Việt Nam

Quan sát người khác để soi lại bản thân — thấy điều tốt thì học, thấy điều xấu thì tránh.

TIN TỨC NỔI BẬT

1

Claude Code vs GitHub Copilot 2026: SWE-bench, Giá [Đã thử nghiệm]

Claude Code vs GitHub Copilot 2026: SWE-bench, Pricing [Tested]

tech-insider.org

[Đọc bài viết →](#)

Trong một so sánh gần đây, Claude Code và GitHub Copilot đã được thử nghiệm trong các chuẩn mực kỹ thuật phần mềm (SWE) khác nhau. Kết quả cho thấy Claude Code vượt trội so với GitHub Copilot ở nhiều lĩnh vực, bao gồm hoàn thiện mã và chất lượng mã. Khả năng hiểu và tạo mã của Claude Code được tìm thấy là chính xác và hiệu quả hơn. Về giá cả, cả Claude Code và GitHub Copilot đều cung cấp các mô hình dựa trên đăng ký. Tuy nhiên, cấu trúc giá cả khác nhau. Giá của GitHub Copilot bắt đầu từ 10 đô la mỗi tháng, với tùy chọn thử nghiệm miễn phí. Giá của Claude Code không được nêu rõ trong bài viết, nhưng nó được đề cập rằng công ty cung cấp một kế hoạch miễn phí, cũng như một tùy chọn đăng ký trả phí. Các thử nghiệm SWE-bench được sử dụng để so sánh hai công cụ mã hóa được hỗ trợ bởi AI này không được chỉ định trong bài viết. Tuy nhiên, kết quả cho thấy Claude Code có thể là một lựa chọn khả thi hơn cho các nhà phát triển (developer) tìm kiếm khả năng hoàn thiện và tạo mã chất lượng cao.

2

Lớp lệnh AI Multi-Agent cho kho hàng: Nâng tầm vận hành và thông minh chuỗi cung ứng

Multi-Agent Warehouse AI Command Layer Enables Operational Excellence and Supply Chain Intelligence

NVIDIA Developer

[Đọc bài viết →](#)

NVIDIA đã giới thiệu Lớp lệnh AI Nhà kho Đa tác nhân, được thiết kế để nâng cao hiệu quả hoạt động và trí tuệ chuỗi cung ứng trong các nhà kho. Giải pháp đổi mới này tận dụng AI để tối ưu hóa hoạt động nhà kho, tự động hóa quy trình và cải thiện việc ra quyết định. Lớp lệnh AI Nhà kho Đa tác nhân được xây dựng trên nền tảng Omniverse của NVIDIA, cho phép tạo ra các bản sao kỹ thuật số tương tác và thực tế cao của các nhà kho. Bằng cách sử dụng công nghệ này, các nhà điều hành nhà kho có thể có được thông tin chi tiết theo thời gian thực về hoạt động của họ, cho phép họ xác định các điểm nghẽn và tối ưu hóa quy trình làm việc. Lớp lệnh AI cũng có thể dự đoán và ngăn chặn các vấn đề tiềm ẩn, chẳng hạn như sự cố thiết bị hoặc thiếu hàng tồn kho, cho phép bảo trì và quản lý hàng tồn kho chủ động. Lớp lệnh AI Nhà kho Đa tác nhân dự kiến sẽ mang lại lợi ích đáng kể cho hoạt động nhà kho, bao gồm năng suất lao động cải thiện, chi phí giảm và khả năng hiển thị chuỗi cung ứng được nâng cao. Bằng cách khai thác sức mạnh của AI và bản sao kỹ thuật số, các nhà điều hành nhà kho có thể đưa ra quyết định dựa trên dữ liệu và thúc đẩy sự xuất sắc hoạt động.

Agoda mã nguồn mở APIAgent: Biến mọi REST/GraphQL API thành MCP Server không cần code

3

Agoda Open Sources APIAgent to Convert Any REST or GraphQL API into an MCP Server with Zero Code

MarkTechPost

[Đọc bài viết →](#)

Agoda đã mở mã nguồn APIAgent, một công cụ cho phép các nhà phát triển chuyển đổi bất kỳ API REST hoặc GraphQL nào thành máy chủ nền tảng Container Microservices (MCP) mà không cần viết mã. Đổi mới này nhằm mục đích đơn giản hóa quá trình tích hợp API với môi trường MCP, giúp nó trở nên dễ tiếp cận hơn với nhiều nhà phát triển. APIAgent sử dụng một thuật toán độc quyền để phân tích cấu trúc của API và chuyển đổi nó thành định dạng MCP tương thích. Công cụ này hỗ trợ nhiều ngôn ngữ lập trình và framework khác nhau, bao gồm Java, Python và Node.js. Bằng cách tận dụng APIAgent, các nhà phát triển có thể nhanh chóng và dễ dàng tích hợp API của họ với môi trường MCP, giúp tối ưu hóa quy trình phát triển của họ. Việc mở mã nguồn APIAgent dự kiến sẽ mang lại lợi ích cho cộng đồng nhà phát triển bằng cách cung cấp một giải pháp tích hợp API miễn phí và mã nguồn mở. Động thái này cũng phù hợp với mục tiêu của Agoda trong việc thúc đẩy đổi mới và hợp tác trong ngành công nghệ. Với APIAgent, các nhà phát triển hiện có thể tập trung vào việc xây dựng

và triển khai ứng dụng của họ mà không cần phải đối mặt với sự phức tạp của tích hợp API.

4

Đừng làm 'meat proxy'

Don't be a meat proxy

Simon Willison

[Đọc bài viết →](#)

Một thuật ngữ gần đây được Niklas Gruhn đặt ra, "meat proxy", ám chỉ những cá nhân vô tư sao chép và dán nội dung được tạo ra bởi AI vào cho người khác. Hiện tượng này xảy ra khi mọi người đưa ra lời nhắc cho các hệ thống AI nhưng không đánh giá một cách nghiêm túc đầu ra, thay vào đó chỉ đơn giản chuyển tiếp nó đến đồng nghiệp của họ. Tác giả nhấn mạnh tầm quan trọng của việc đóng một vai trò tích cực trong quá trình này bằng cách đọc, hiểu, xác thực và sau đó viết một phản hồi bằng lời của riêng mình. Sự nỗ lực bổ sung này không chỉ thể hiện giá trị của suy nghĩ và quan điểm riêng của một người mà còn cung cấp một quan điểm độc đáo mà các hệ thống AI có thể không thể sao chép. Bằng cách làm như vậy, các cá nhân có thể thêm giá trị vào cuộc trò chuyện và tránh được việc bị coi là những "meat proxy" đơn thuần chỉ dựa vào nội dung được tạo ra bởi AI.

5

Tự động hóa với tốc độ của Swamp

Automation at the speed of Swamp

Changelog

[Đọc bài viết →](#)

Lãnh đạo ngành công nghệ Adam Jacob, người sáng lập System Initiative và tạo ra Swamp, chia sẻ những nhận xét của mình về cách các tác nhân AI đang cách mạng hóa việc phát triển phần mềm. Mặc dù giảm quy mô đội ngũ 18 người xuống chỉ còn năm người, Jacob vẫn có thể xuất bản Swamp 900 lần trong bốn tuần, nhấn mạnh hiệu quả và tốc độ mà AI có thể mang lại cho quá trình này. Ông nhấn mạnh tầm quan trọng của kiểm thử chấp nhận người dùng và kiến trúc phần mềm, gợi ý rằng những khía cạnh này hiện nay quan trọng hơn kỹ năng lập trình. Jacob cũng trình diễn khả năng của Swamp bằng cách sử dụng nó để tự động hóa các nhiệm vụ trên hộp Proxmox của mình, □ tiềm năng của công nghệ này trong việc viết tự động hóa của chính nó. Ngoài ra, ông giải thích tại sao ông sẽ không chấp nhận yêu cầu kéo (pull requests) đối với Swamp, cho thấy sự chuyển dịch hướng tới các quy trình phát triển tự chủ hơn.

6

Sau quý kinh doanh kỷ lục, CEO Palantir Alex Karp gọi ngành AI là 'Marxist'

After killer quarter, Palantir CEO Alex Karp calls AI industry 'Marxist'

TechCrunch AI

[Đọc bài viết →](#)

Giám đốc điều hành Palantir Alex Karp đã bày tỏ mối quan ngại về sự ảnh hưởng ngày càng tăng của các phòng thí nghiệm AI tiên phong, so sánh họ với các nhà tư bản chủ nghĩa Mác, những người nhằm mục đích chiếm đoạt phương tiện sản xuất từ các đối tác của họ. Trong một bức thư cổ đông hàng quý, Karp lưu ý rằng doanh thu kỷ lục của Palantir là 1,9 tỷ USD, tăng 93% so với năm trước, đã được đạt được mặc dù sự tăng trưởng nhanh chóng của các phòng thí nghiệm AI. Trong một cuộc gọi hội nghị với các nhà phân tích, Karp đã làm rõ thêm về ẩn dụ của mình, đề xuất rằng các công ty hợp tác với các phòng thí nghiệm AI cơ bản đang đầu hàng quyền kiểm soát dữ liệu và tài sản trí tuệ của họ. Ông lập luận rằng xu hướng này được thúc đẩy bởi một cảm giác về sự ưu □ đạo đức giữa các nhà lãnh đạo phòng thí nghiệm AI, những người nhằm mục đích thuộc địa hóa các doanh nghiệp và xây dựng các doanh nghiệp cạnh tranh không cần các đối tác của họ. Các bình luận của Karp phản ánh một mối quan ngại ngày càng tăng rằng các phòng thí nghiệm AI đang ưu tiên lợi ích của riêng

họ hơn là của các đối tác của họ, và rằng xu hướng này có thể có những tác động đáng kể đối với tương lai của ngành công nghiệp.

7

Cách đánh giá Sourcegraph trên codebase của bạn

How to evaluate Sourcegraph on your own codebase

Sourcegraph Blog [Đọc bài viết →](#)

Đánh giá hiệu quả của Sourcegraph trên một cơ sở mã yêu cầu một cách tiếp cận tinh tế. Đơn giản chỉ đo lường chất lượng truy xuất hoặc hoàn thành nhiệm vụ có thể không cung cấp một bức tranh hoàn chỉnh. Các tác giả của bài viết này cho rằng độ khó truy xuất là một thuộc tính của cả cơ sở mã và nhiệm vụ, và các chỉ số đo lường khác nhau đo các khía cạnh khác nhau của hệ thống. Họ trình bày một nghiên cứu trường hợp nơi việc truy xuất mã có cấu trúc của Sourcegraph đã cải thiện F1 và độ nhớ hồi tại mức tệp, nhưng có ít tác động đến phần thưởng hoàn thành nhiệm vụ tổng hợp. Tiết kiệm chi phí cũng thay đổi tùy thuộc vào nhiệm vụ và liệu tác nhân cơ sở có thể nhanh chóng xác định vị trí công việc hay không. Để đánh giá chính xác Sourcegraph, điều quan trọng là phải phân loại nhiệm vụ theo độ khó xác định vị trí và báo cáo công việc xác định vị trí dễ dàng riêng biệt với các nhiệm vụ yêu cầu khám phá thực sự. Cách tiếp cận này sẽ giúp xác định liệu hệ thống truy xuất của Sourcegraph có lợi cho công việc cụ thể mà các kỹ sư và tác nhân của bạn gặp khó khăn trong việc xác định vị trí.

8

Sau Orthogonality: Agency đạo đức đức hạnh và AI Alignment

After Orthogonality: Virtue-Ethical Agency and AI Alignment

The Gradient [Đọc bài viết →](#)

Trong bài viết này, tác giả cho rằng con người hợp lý và trí tuệ nhân tạo (AI) không nên tập trung vào mục tiêu. Thay vào đó, họ nên căn chỉnh hành động của mình với các thực hành, là mạng lưới các hành động, khuynh hướng, tiêu chí đánh giá và tài nguyên cấu trúc và thúc đẩy bản thân. Tác giả khẳng định rằng cách tiếp cận này là điều cần thiết để tạo ra các AI có thể thực sự hỗ trợ và cộng tác với con người. Bài viết cũng khám phá khái niệm eudaimonia, hay sự thịnh vượng con người tích cực và hợp lý, và cho rằng nó gợi ý một hình thức hoạt động hợp lý mà không có sự phân biệt nghiêm ngặt giữa phương tiện và mục đích. Tác giả đề xuất rằng hình thức hợp lý này, được gọi là hợp lý

eudaimonic, là một khuôn khổ hữu ích cho cơ quan và giá trị của AI được căn chỉnh với con người, vì nó ổn định và an toàn hơn hợp lý truyền thống theo hệ quả. Tác giả gợi ý rằng hợp lý eudaimonic là một hình thức cơ quan tự nhiên và nó có hiệu quả trong việc căn chỉnh AI với sự thịnh vượng của con người.

TIPS & TRICKS CHO DEV

Tự động hóa test case

Vấn đề: Viết test case thủ công tốn thời gian và dễ xảy ra lỗi.

Cách làm: Sử dụng AI tools như Testim.io để tự động tạo test case. Ví dụ, prompt "Create test case for login functionality" để tạo test case cho tính năng đăng nhập.

Đánh giá: Hiệu quả cao, tiết kiệm thời gian, nhưng cần kiểm tra kết quả.

Code review tự động

Vấn đề: Code review thủ công tốn thời gian và dễ bỏ lỡ lỗi.

Cách làm: Sử dụng công cụ như SonarQube để tự động kiểm tra code. Ví dụ, lệnh "sonar-scanner" để quét code và tìm lỗi.

Đánh giá: Hiệu quả cao, giúp tìm lỗi sớm, nhưng cần cấu hình chính xác.

Tự động hóa QA automation

Vấn đề: Viết script tự động hóa QA tốn thời gian và khó bảo trì.

Cách làm: Sử dụng AI tools như Selenium để tự động tạo script. Ví dụ, prompt "Create script for automated login test" để tạo script cho thử nghiệm đăng nhập tự động.

Đánh giá: Hiệu quả cao, tiết kiệm thời gian, nhưng cần kiểm tra kết quả và bảo trì thường xuyên.

BÀI HỌC AI HÔM NAY CHO DEV

1. Tối ưu chi phí & hiệu năng LLM

2. Việc tối ưu chi phí và hiệu năng của Large Language Model (LLM) là rất quan trọng để đảm bảo ứng dụng AI hoạt động hiệu quả và tiết kiệm. Dev cần biết cách tối ưu hóa LLM để giảm thiểu chi phí tính toán và tăng tốc độ xử lý. Điều này giúp cải thiện trải nghiệm người dùng và giảm gánh nặng cho hệ thống.

3. Ví dụ, có thể sử dụng kỹ thuật fine-tuning và LoRA (Low-Rank Adaptation) để tối ưu hóa LLM cho use case cụ thể, giảm thiểu số lượng tham số và tăng tốc độ huấn luyện.

4. Tip hoặc bước tiếp theo: Hãy thử sử dụng thư viện như Hugging Face Transformers để triển khai và tối ưu hóa LLM trong dự án của bạn, đồng thời theo dõi các chỉ số hiệu năng và chi phí để điều chỉnh và cải thiện.

Luôn đi đầu trong thế giới AI! · Stay ahead in AI!

Nguồn: Google News · Groq AI